REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	February 23,2000	Final Report 9/23/96- 6/22/99

4. TITLE AND SUBTITLE
Multiagent Systems in Logistics Environment

5. FUNDING NUMBERS
DAAH04-96-1-0464

6. AUTHOR(S)
Soundar R.T. Kumara

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
The Pennsylvania State University
310 Leonhard Bldg.,University Park, PA 16802

8. PERFORMING ORGANIZATION REPORT NUMBER
The Pennsylvania State University

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U. S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
ARO 36389.6-MA

11. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.

12 a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12 b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

A procurement problem for a manufacturing company is to decide the suppliers and manufacturing locations from a list of prospective candidates for the manufacture of products that will minimize the cost added to the price of the products. This is a typical logistics problem. Under some assumptions, this problem can be solved through negotiation in a forum of candidate transportation companies. From this perspective, we can model each transportation company as comprising of a master agent and several slave agents. A master agent negotiates for tasks with the master agents of other companies and assigns tasks to its own vehicles. A slave agent represents a vehicle which generates robust vehicle routes, and cost estimation for an assigned task by collaborating with other vehicles. We develop a theoretical information infrastructure for solving this problem using multiagents, game theory and stochastic programming. We apply this formalism to a logistics problem posed by USALIA and build an implementation platform.

14. SUBJECT TERMS

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OR REPORT	18. SECURITY CLASSIFICATION ON THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

DTIC QUALITY INSPECTED 4

Standard Form 298 (Rev.2-89)
Prescribed by ANSI Std. 239-18
298-102

1. List of Manuscripts Submitted

- Satapathy, G., and Kumara, S. R. T., *Collaborative Multiagent based information Infrastructure for Transportation Problem Solving*, to appear in the Journal of Information System Frontiers, February 1999.
- Satapathy, G., and Kumara, S. R. T., Negotiation for Transportation Tasks with Stochastic Payoffs, To appear in the the Journal of Computer In Industry, 2000.
- Satapathy, G., and Kumara, S. R. T., *Negotiation for Transportation Tasks with Stochastic Payoffs*, In the Workshop of the 3rd International Conference on Autonomous Agents (Agent'99), Agents for Electronic Commerce and Managing the Internet-Enabled Supply Chain, Seattle, Washington, May 1999. Extended version accepted in the Journal of Computer In Industry, 2000.
- Satapathy, G., and Kumara, S. R. T., *The object oriented design based agent modeling*, In the Proceedings of the Fourth International Conference on Practical Applications of Agents and Multiagents, London, UK, 1999.
- Satapathy, G., Kumara, S. R. T., and Moore, L. M., *Software Agents in Logistics*, in Encyclopedia of Library and Information Science, Volume 67, Allen Kent (ed.), Marcel Dekker, NY, 2000 (in press).
- Reyns, B., Smith, G., Satapathy, G., Kumara, S. R. T., and Hummel, J., *Distributed Intelligent Agents for Logistics: an update*, In the Proceedings of Simulation Interoperability Workshop, Spring 98, Simulation Interoperability Standards Organization, 98S-SIW-203, Orlando, FL, March 1998.
- Satapathy, G., Kumara, S. R. T., and Moore, L. M., *Distributed Intelligent Architecture in Logistics (DIAL)*, International Journal of Expert Systems with Applications, Vol. 14, pp409-424, 1998.

2. Scientific Personnel Supported by the project and Honors /Awards/Degrees received during this period

Nina M. Berry
Glen Smith
Tim Thomas
Bryan
Goutam Satapathy
T.R. Seshadri
Yonghan Lee
Lin Yeh
Soundar R.T. Kumara

Degrees Awarded (only those theses which are directly related to the research proposal are reported):

1. Nina M. Berry: Ph.D., in Industrial Engineering (African American Women)

Title: Reagere : A hybrid multi-agent architecture for manufacturing, The Pennsylvania State University, 1997.

Currently employed by Sandia National Laboratories, Livermore, CA.

2. Goutam Satapathy: Ph.D., in Industrial Engineering

Title: Distributed and Collaborative Logistics Planning and Replanning under Uncertainty: A Multiagent Approach, The Pennsylvania State University, 1999.

Currently Employed by Intelligent Automation Incorporated, Virginia.

3. Seshadri, T.R., M.S., in Industrial Engineering

Title: An application of software agents to internet based on-line customer order processing, The Pennsylvania State University, 1998.

Currently Employed by Bell Core.

4. Lin Yeh , M.S., in Industrial Engineering
Title: An agent-based framework for business process integration with application to complex order
Management, The Pennsylvania State University, 1999.

Currently employed by a start-up company in Silicon Valley.

3. Scientific Progress and accomplishments

In a supply chain, a manufacturing company on certain occasions needs to decide the suppliers from whom the raw materials must be obtained, and the manufacturing sites where the final products must be manufactured, to cater to the needs of specific types of customers. This decision-making problem is termed as *procurement problem.* Under certain assumptions, such as cost and quality of the raw materials and the final product are independent of the suppliers and the manufacturing sites, the solution of the procurement problem mainly consists of the least cost means of transportation between the suppliers and the manufacturing sites, and between the manufacturing sites and the customer locations. In this research, we address how to solve a procurement problem when the manufacturing company seeks freight companies as the means of third party transportation. The fundamental issues that are addressed in this work are: (1) minimization of bid (transportation cost) for the manufacturing company, (2) maximization of payoff to the freight companies, (3) cost-optimal and cost-reliable estimation of the task-to-vehicle assignments by the freight companies, and (4) reliable fulfillment of the orders by the drivers of the vehicles. Based on these issues, we define three research objectives. (1) To develop a distributed problem-solving model, define the decision-making entities and formalize their interactions. (2) To model a transportation task selection and distribution procedure that minimizes the transportation cost to the manufacturing company and increases the payoff to the freight companies, compared to the current practice. (3) To develop a distributed collaborative algorithm to generate

reliable vehicle routes under the assumption that there exists an opportunity for the drivers of the vehicles to transfer goods among each other.

The problem-solving model developed in this work is a multi-agent computational model, in which each freight company is comprised of a master agent and several slave agents. A master agent of a freight company represents a computational entity, which negotiates with the master agents of other freight companies for transportation tasks, and assigns sub-tasks to the drivers of its vehicles. A salve agent is a computational decision making unit on the behalf of a driver of a vehicle that generates robust routes for the driver of that vehicle. The agents are modeled using a Belief-Desire-Intention (BDI) logical model and interact using (modified) KQML based messages.

The task selection and distribution procedure consists of a discounted bargaining game model with alternating offers which is similar to a multi-commodity reverse English auction. We conduct a simulation to study how masters of the freight companies can improve their payoffs by adopting (1) strategic bargaining procedure and (2) deploying different (complete) real-time cost information acquisition systems. The primary result is that masters do not always require expensive cost information acquisition systems, if they can predict the cost of the tasks accurately and follow a strategic bargaining procedure.

A distributed Collaborative Robust Vehicle Route (CRVR) algorithm is presented for reliable and optimal cost estimation of the tasks in real-time. The CRVR algorithm solves a single vehicle stochastic pickup and delivery problem (PDP). The important contribution of this algorithm is that it uses the driver's local behavior along with travel time and service time acquired in real-time to determine the bottlenecks on a feasible route, and announce collaboration requests to delegate certain orders created to eliminate the bottlenecks. The algorithm presented solves a stochastic PDP with ten transportation orders within reasonable time. Furthermore, the results also report that the cost of the tasks can reduce due to collaboration under certain circumstances.

The results of this research can be extended to build an integrated supply chain infrastructure in any logistics scenario, in the e-commerce world. Real-time sensors can be incorporated with the problem-solving model to implement real-world system.

The major new accomplishments are:

1. A multiagent based problem solving model for logistics
2. An ontology for logistics
3. A Game theory based algorithm for collaboration
4. A stochastic programming based collaborative robust vehicle route generation model.

Multiagents in Logistics Environment

Army Research Office

Grant# DAAH-04-96-1-0464

9/23/96 – 6/22/99

Final Report

Soundar R. T. Kumara

Intelligent Design and Diagnostics Research Laboratory
Department of Industrial and Manufacturing Engineering
The Pennsylvania State University
University Park, PA 16802, USA

kumara@marie.iddr.ie.psu.edu
http://www.iddr.ie.psu.edu
Ph: (814) 863 2359, Fax: (814) 863 4745

Multiagents in Logistics Environment: Final Report

Soundar R. T. Kumara
Intelligent Design and Diagnostics Research Laboratory
Department of Industrial and Manufacturing Engineering
The Pennsylvania State University
University Park, PA 16802, USA

kumara@marie.iddr.ie.psu.edu
http://www.iddr.ie.psu.edu
Ph: (814) 863 2359, Fax: (814) 863 4745

Foreword

In the Multiagents in Logistics Project funded by the ARO we have performed two research tasks. These are:

1. a theoretical investigation into the application of multi-agents in logistics, with specific reference to the procurement problem in which we have developed a stochastic programming and game theory based approach for solving the problem, and
2. a practical implementation of multi-agents to a problem of logistics systems integration posed by the United States Logistics Integration Agency at New Cumberland, PA. We have built and deployed a system, Distributed Intelligent Agents in Logistics(DIAL).

In this final report we give a brief, yet a technical overview of the above two. In the first part of the document we discuss the Procurement Problem and in the second part.

Table of Contents

PART-I: Multiagents in Procurement Problem Solving

1.0 Procurement Problem

A manufacturing company, say GE, has refrigerator manufacturing plants at three locations:

1. Schenectady, NY
2. Erie, PA
3. York, PA

Based on the demand by a certain distributor at Pittsburgh, PA (who is the customer to GE) GE wants to manufacture 50 refrigerators that can be delivered at the distributor by July 20, 1999 (the current date being July 13, 1999). Assume that the refrigerators manufactured at these three locations meet the distributor's quality criteria and the manufacturing cost at each of these plants is same (or comparable). The decision of where to manufacture these refrigerators depends on (1) the final cost (sum of the parts cost, manufacturing cost and transportation cost), and (2) the quantity, which can be manufactured at each plant that meets the delivery time requirement. Assume that the plants (after looking at their current schedule) present the following requirement.

1. **Schenectady, NY:** Refrigerators in the multiples of 10 up to 40 can be manufactured that will meet the delivery requirement given that following parts are delivered by this time
   a. Compressors for all refrigerators are delivered by July 16, 1999
   b. Defrosters for all refrigerators are delivered by July 17, 1999

2. **Erie, PA:** Refrigerators in the multiples of 5 up to 35 can be manufactured that will meet the delivery requirement given that following parts are delivered by this time
   c. Compressors for all refrigerators are delivered by July 18, 1999
   d. Defrosters for all refrigerators are delivered by July 18, 1999

3. **York, PA:** Refrigerators in the multiples of 15 up to 45 can be manufactured that will meet the delivery requirement given that following parts are delivered by this time
   e. Compressors for all refrigerators are delivered by July 15, 1999
   f. Defrosters for all refrigerators are delivered by July 16, 1999

Assume that all other parts are manufactured locally by GE or are available in stock. The compressor and defroster suppliers of GE are located at the following locations:

1. Albany, NY
2. Scranton, PA
3. Cleveland, OH

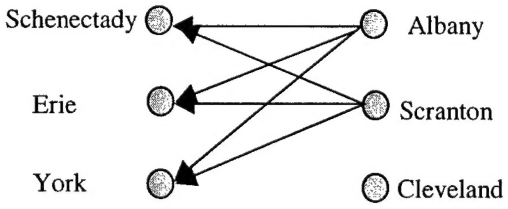
Assume that the suppliers provide following information:

1. **Albany, NY:** 40 defrosters will be available for pickup by 15 July 1999. It cannot supply compressors on time.

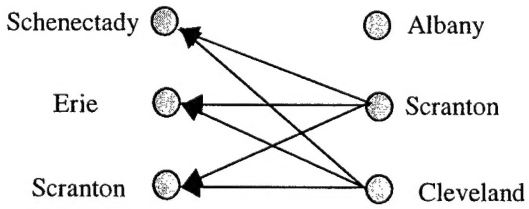
2. **Scranton, PA:** 20 compressors will be available for pickup by 14 July 1999 and 40 defrosters will be available for pickup by 15 July 1999

3. **Cleveland, OH:** 30 compressors will be available for pickup by 15 July 1999. It cannot supply defrosters on time

Assume that the cost of the compressors and the defrosters are same (or comparable) from all suppliers. Also all suppliers meet the quality standards. Now the procurement problem for GE consists of: ***from where should it obtain the compressors and the defrosters, and where should those parts be delivered for the manufacture of the refrigerators so that refrigerators are delivered at the Pittsburgh distributor by July 20 and the cost is remains minimum.*** Since the cost of the parts is same at all suppliers, the manufacturing cost is same at all plants, and the quality standards are met, *the problem reduces to how cheaply can GE transport the parts from suppliers to the plants that meets the time requirements.*



Defroster supply lines



Compressor supply lines

It must be noted that these two supply lines are dependent. That means, the compressors cannot be delivered to a plant where the defrosters are not supplied. But in our formalism, they are independent. This can be incorporated by eliminating some of the *agreements* (explained in the ontology) that violates this dependence.

Assume that GE has contracts with the following freight companies who would transport any parts required by GE plants.

1. Consolidated Freightways, Inc. (CF)
2. ABF Freight System, Inc (ABF)
3. RPS, Inc (RPS)

The traditional (contemporary) method is to request for bids from these freight companies for each *supply line*. The GE will determine (by solving a simple optimization problem) the best task (i.e. supply line) distribution to these freight companies. However, due to *one time submission* of bids, the following drawbacks are experienced:

1. GE sometimes pays more for the transportation than the correct or market-accurate cost
2. GE although sometimes pays less, it does not necessarily results in timely delivery by the freight company. (This happens when a freight company has bidden over-optimistically)
3. A freight company is often charged with penalties for lack of timely delivery.
4. A freight company sometimes bear transportation cost more than the bid it has offered for transportation (thus it incurs loss)
5. A freight company sometimes bids pessimistically. Thereby, it looses to other freight companies during the bidding process.

In order to remove these drawbacks using the emerging information infrastructure such as web, internet, EDI, GIS, GPS, cellular/satellite communication, we propose *an auction based mechanism* to solve the procurement problem. In this mechanism, the freight companies negotiate to distribute the tasks (thereby select the supplier and the manufacturing plant for each part) using the cost information gathered in real-time. **This is the heart of the problem that we are addressing.**

In order to distribute the tasks (i.e. supply lines), the freight companies require a set of agreements to bargain with. Now using our ***product**** *notations described in our formalism*, we combine 5 refrigerator as one product†. That means, distributor requires 10 (10x5) different products. Similarly, Schenectady, NY can manufacture 8 (8x5) products, Erie, PA can manufacture 7 (7x5) products and so on. Since each refrigerator requires one compressor and one defroster, we extend the notations. Schenectady, NY requires 8 parts ($c_1 - c_8$) [c_1 to c_8 mathematically denote different parts, but they are of compressor part type] and 8 parts ($d_1 - d_8$) [d_1 to d_8 mathematically denote different parts, but they are of defroster part type]. Using these notations, we formulate the set of agreements, **X.**

2.0 Breif Overview of the Approach

The procurement problem consists of deciding the plants where the final product i.e. the refrigerators can be manufactured, and the supplier who can supply the part compressors and defrosters to those plants. Assuming the plants and the suppliers of each part is determined, the second step is to decide how to distribute the transportation orders of the parts to the freight companies so that the total price (bids) paid by the manufacturing company is minimized. Assuming that a freight company, say ABF Freightways Inc, is awarded with one or more transportation order(s), the third and the final step is to decide

* A product to transported is a part in this case i.e. compressors and defrosters. These parts are nothing but products from the freight companies point of view. That is why, we use the term "product" in general.

† The number 5 can be derived by determining GCF of the quantities required, quantities that can be manufactured and supplied

which vehicle[‡] to assign with the transportation orders so that the freight company incurs minimum cost and maximum profit. These three steps depend on each other. The decision at each of these steps cannot be taken independently or sequentially. Therefore, we need to address the concurrent problem solving process, assuming that the auction based mechanism is the solution framework.

The problem solving process consists of the manufacturing company throwing the problem to a group of freight companies who will enter into an auction based negotiation/bargaining to select and distribute the tasks (thereby select the supplier and the manufacturing plant for each part). In order to offer bids for tasks, a freight company must know the expected payoff from each of the task. Determining the expected payoff from each task is tantamount to determining the expected cost of each task. Therefore, in the problem solving process, the freight company (i.e. the computational entity called master agent) presents the sub-tasks to different vehicles (i.e. a computational entity called slave agent which represents the drivers of a vehicle) in real-time. The slaves in turn collaborate with each other to determine a least cost reliable path, and replay back with the expected cost of fulfilling the task. Masters use this information for bidding until no master can reduce their bid any further based on the cost and reliability information provided by their slaves (and the company's strategic business practice). If the result of bargaining dictates that a freight company has been awarded to complete a task, then the freight company assign the sub-task to the vehicles which corresponds to one set of sub-tasks presented to the slaves/vehicles during the bidding cycle. The drivers of those vehicles execute the plans to fulfill the sub-tasks, which where generated by the slaves during the collaboration phase. During the execution, if the plan of a vehicle fails then the remaining tasks are delegated to the driver of another vehicle, according to the contingency plan that was generated during collaboration. It is possible that if we use accurate sensor information, and sensor reliability data, then we can avoid re-planning by predicting the failure apriori and drawing collaborative contingency plans.

3.0 Ontology

The agents use KQML based messages for communication. This has the following structure:

```
(<performative>   :sender < word >
                  :receiver < word >
                  :reply-with < word >
                  :in-reply-to < word >
                  :language < word >
                  :ontology < word >
                  :illocutionary_act < AnIllocutionaryAct >
                  :content < expression >)
```

[‡] We assume a driver as the owner of a vehicle and hence a vehicle is already assigned with a driver

The <expression> in the content uses a language which is a derivative of predicate logic and we call it as Logistics Language (LogL). The BNF grammar of LogL is:

<predicate>({<type> <r|w|rw|> <arg> <val> })

<predicate> is similar to a directive of a message. <arg> are the arguments required for either understanding the message or executing the directive. <type> indicates the structure or class (similar to OOP) of the argument. The types are predefined and part of ontology. <arg> can be <word> or <predicate> itself. <val> is the value of a <arg> expressed in the format of its <type> i.e. class defined. The following is a minimal set of the terms used by the master and slave agents to solve procurement problem:

N	List of nodes. A node indicate a location e.g. city name
J	Set of part indices
O	List of orders. An order, O is specified by a pickup node, pickup time window, a delivery node, delivery time window, and the part index.
T	List of tasks. A task, T is a list of orders that observes two order non-conflicting constraints (not explained here)
P	Set of freight companies
V	A group of vehicles/slaves of a freight company
b	Bid for transportation of a part
x	An agreement. An agreement is the list of parts such that each part is associated with freight company who is transporting the part, the task that the freight company is following, and the bid it has offered for transportation
X	The set of agreements
CMNP	A term that encapsulates bargaining rules (not explained here)
p	Probability of completing a task
c	Cost of completing a task
SCENE_ID	A scenario id. An expression used by master while presenting tasks to the slaves

The following table is the set of predicates and their <args>s used by master and slaves to communicate with each other.

Sl. No.	**Syntax**	**From (speaker)**	**To (receiver)**	**Implications or Outcome or Action taken by the receiver**
1	*Performative*: ask-one *Predicate:* NegotiateTasks	User (Mfg. Company)	Masters	The arguments of NegotiateTasks contain CMNP, **T, X, P**. (Task T is output) The masters unpon receiving this message enter a negotiation phase
2	*Performative*: subscribe *Predicate*: KQML, GetAgreement	Master-I Master-II	Master-II Master-I	Masters subscribe to each other for receiving offers and counter-offers

3	*Performative*: ask-one *Predicate*: GetCostnProb	Master-I	Slaves	Master presents a Task to a set of slaves. The arguments of GetCostnProb is a task T, **V**, c, p, SCENE_ID (c and p are ouputs). **V** is the set of slaves/vehicles with whom it can collaborate.
4	*Performative*: ask-one *Predicate*: GetCostnProb	Slave-I Slave-II	Slave-II Slave-I	Using the planning algorithm, a slave (I) asks another slave (II) to complete a part of task. The argument of GetCostnProb contains order O, null, c, p, SCENE_ID. null implies that the slave to whom it request cannot collaborate with others. Slave (I) can also receive similar request from other slaves (II)
5	*Performative*: tell *Predicate*: CostnProb	Slave-II Slave-I	Slave-I Slave-II	The requested slaves (II) submit their estimated cost and probability. The arguments of CostnProb contains same with c and p completed.
6	*Performative*: insert delete *Predicate*: ToBeAssigned	Slave-I Slave-II	Slave-II Slave-I	The slave (I) requests one slave (II) to know that it will be assigned with an order in future. The arguments of ToBeAssigned contains a task T and SCENE_ID.
7	*Performative*: tell *Predicate*: CostnProb	Slave-I	Master-I	Slave (I) reports the cost c and probability p of the collaborative plan to the master.
8	*Performative*: tell *Predicate*: AnAgreement	Master-I Master-II	Master-II Master-I	Masters use the information provided by slaves to determine the best agreement and offer and counter-offer each other. It must be noted that the steps 3-7 may be repeated during each offer and counter-offer.
9	*Performative*: unsubscribe *Predicate*: KQML, GetAgreement	Master-I Master-II	Master-II Master-I	Masters unsubscribe when they finalize on an agreement.
10	*Performative*: tell	Master-I	User	The masters tell the users their individual tasks based on the

	Predicate: ATask			final agreement.
11	*Performative*: achieve *Predicate*: Atask	User	Master-I	User (if approves, else reformulate the arguments of NegotiateTasks) directs the masters to execute the tasks
12	*Performative*: achieve *Predicate*: ATask	Master-I	Slave-I	Each master then directs the salves to execute the tasks that they had presented during repetition of step 3-7
13	*Performative*: achieve *Predicate*: APlan	Slave-I	Execution	Slave directs the execution environment to execute a plan determined during step 3-7. The arguments of APlan is the departure time at each node of a sequence of nodes.
14	*Performative*: tell *Predicate*: ATask	Slave-I	Master-I	Slave replies to the master that it has committed to the task
15	*Performative*: tell *Predicate*: ArrivalTime	Execution	Slave-I	The execution environment keeps reporting the arrival times of the vehicle. If the slave determines that it needs to stop execution (to declare failure or delegate tasks) it sends unachieve message similar to step 13.
16	*Performative*: achieve *Predicate*: ATask	Slave-I	Slave-II	If the salve needs to delegate a task, it will direct the task to the slave to whom it had sent a message earlier (step 6)
17	*Performative*: tell untell *Predicate*: ATask	Slave-II	Slave-I	Slave II replies that it commits to the task. Slave II can also reply telling it cannot fulfil the task (untell)
18	*Performative*: untell *Predicate*: ATask	Slave-I	Master-I	If a slave observes that it cannot fulfill a task assigned by its master, then it replies back to message in step 12, by declaring failure of an assigned task.
19	*Performative*:	Master-I	Master-II	A master tries to subcontract the

	ask-one *Predicate*: NegotiateTasks			failed task to another master. Therefore, this message is similar to the message in step 1. After several steps similar steps 1-10, the Master-I directs the Master II to execute the failed task, which is similar to step 11 (user directing the Master I)
20	*Performative*: untell *Predicate*: ATask	Master-I	User	Master I, which has a failed task, informs the user of the failure. It does not de-commit. It cannot de-comit by the rules of the business. It just informs that another Master II will fulfill the task.
21	*Performative*: tell *Predicate*: ATask	Master-II	User	Master II commits to fulfill the failed task by sending a message to the user. This information is required for user to know that vehicles of another company will be served at a delivery node.

4.0 Bargain Model

Assume there is a set of 10 agreements denoted by $[\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7, \mathbf{x}_8, \mathbf{x}_9, \mathbf{x}_{10}]$. Remember that each agreement specifies the freight company that will be transporting each of the products (i.e. parts), the task (i.e. supply line) chosen by the freight company, and the bid offered by the company. For example, $\mathbf{x}_1$ may be specified as:

10 compressors are to be transported from Scranton, PA to Schenectady, NY by ABF ($200)
10 compressors are to be transported from Scranton, PA to York, PA by CF ($180)
10 compressors are to be transported from Cleveland, PA to Erie, PA by RPS ($170)
20 compressors are to be transported from Cleveland, PA to York by CF ($420)
10 defrosters are to be transported from Albany, NY to Schenectady, NY by RPS ($140)
20 defrosters are to be transported from Albany, NY to York by ABF ($290)
10 defrosters are to be transported from Scranton, PA to York by ABF ($145)
10 defrosters are to be transported from Scranton, PA to to Erie, PA by CF ($130)
Total = $ 1675

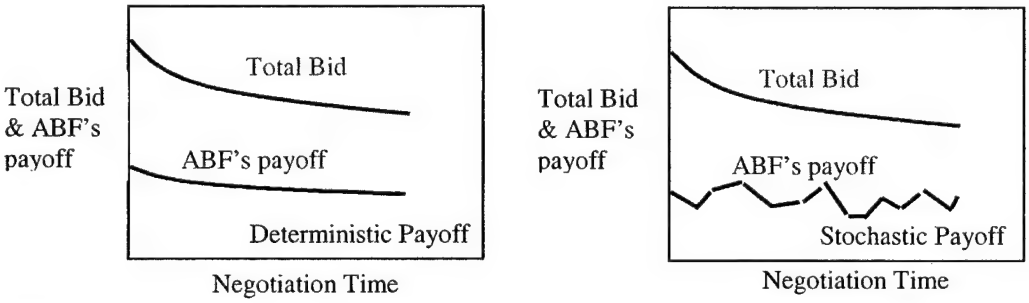
This is a valid agreement that meets the distributor's requirement of 50 refrigerators, the quantity requirement of the plants, and quantity available at the suppliers. Similarly, other agreements can be specified. (*Please note that there are more than 10 agreements for our problem. We are just citing 10 to explain the bargain model*). An agreement say $\mathbf{x}_1$ is agreeable by all the freight companies, if (1) the freight companies has resources i.e. vehicles/trucks that can meet all the timing requirements specified in $\mathbf{x}_1$ and (2) there does not exist another agreement (e.g. $\mathbf{x}_3$ or $\mathbf{x}_7$) from which they can get more payoff. **If a company decides to counter-offer the currently offered agreement then it must**

observe certain rules. The notions of these rules are consistent with the auction based business practices and also leads to convergence. The rules are:

Rule1: *Customer oriented*: If ABF decides to counter-offer $\mathbf{x}_1$ with $\mathbf{x}_3$ then the total bid value of $\mathbf{x}_3$ should be less than or equal to $\mathbf{x}_1$ (i.e. \$1675). In other words, customer i.e. the manufacturing company must not pay more than the currently prevailing agreement i.e. $\mathbf{x}_1$

Rule2: *Deadlock free*: If ABF decides to counter-offer $\mathbf{x}_1$ with $\mathbf{x}_3$ then the total bid value of $\mathbf{x}_3$ should not be equal to $\mathbf{x}_1$ (i.e. \$1675). Else the companies would enter a deadlock in negotiation.

Rules3: *No enmity*: If ABF decides to counter-offer $\mathbf{x}_1$ with $\mathbf{x}_3$ then the counter-offer $\mathbf{x}_3$ *cannot reduce the bid/payment* to another freight company (e.g. CF, \$730 or RPS, \$310) specified in $\mathbf{x}_1$, while ABF does not get any additional bid/payment from $\mathbf{x}_3$ over $\mathbf{x}_1$ (i.e. \$635). In other words, there does not exist any enmity.



These rules together lead to a property called **monotonic decrement of the total bid-value** of the agreements with negotiation (or time). This property along with the **finite set of agreements**, leads to the convergence. For example, as shown in the figure the total bid value keeps decreasing with the time/negotiation step. That means, the payment that will be made to a freight company (e.g. ABF) by the manufacturing company (e.g. GE) decreases with time, if ABF prolongs the negotiation. So eventually there occurs a stage, when ABF cannot find another agreement left in the set of agreements, from which it can expect a higher payoff. This is also mathematically proven. The agreement, to which all freight companies finalize, is unique, if the expected payoff from each agreement to each freight company is deterministic (Although there is not a unique way to arrive at the final agreement. In terms of game theory, it is called non-unique sequential equilibrium). However, in real world, the expected payoff from each agreement is stochastic. Therefore, we conduct a simulation to study the losses that will occur to freight company due to stochasticity of the expected payoff (payoff is obtained from solving task by using a vehicle route planning algorithm), and thereby improve the company's negotiation strategies (i.e. which agreements to offer and counter-offer)

5.0 Practical Issues

1. How can we incorporate, some of the customer (i.e. manufacturing company imposed) constraints such as convenience to deal with a freight company?
   a. For instance, the freight company CF does not deliver parts at the plant door
   b. For instance, the freight company RPS does not handle parts very well (i.e. breakage)
   c. For instance, the freight company ABF does not deliver parts on time.
   d. For instance, the freight companies CF and ABF require minimum number of parts for transportation to strike a deal (i.e. agreement)

Answers:

a. This can be handled in two ways. First, if GE knows the cost of transporting from the delivery point of CF to its plants, then this cost can be added in the bids of the agreement formalization. That means, if CF offers a bid for $170 for a supply line between Scranton, PA to York, PA, and the cost of transporting parts from the delivery point (near York, PA) to the plant in York, PA is $40, then the actual bid of CF is $210.

   The second way to handle is that CF can sub-contract the transportation of parts from its delivery point to the door of GE's plant to another local freight company, and use that sub-contract cost information in offering or counter-offering.

b. If RPS does not handle some parts very well, then the loss to GE due to the past breakage can be incorporate as a penalty to RPS, while formulating the bids of the agreements. This penalty can be done at a strategic level between the GE and RPS, which is beyond the scope of this approach.

c. Similar to the above approach, the losses due to untimely delivery in past by ABF can be incorporated as penalty to ABF, while formulating the bids of the agreements.

d. The minimum number/weight of parts/goods/product transportation requirement by a freight company can be easily incorporated by eliminating some of the infeasible agreements from the set of agreements at the very outset of negotiation.

PART-II : Distributed Intelligent Agents in Logistics (DIAL)

1 Introduction

The agent in DIAL (Distributed Intelligent Agents in Logistics) is perceived as a logistician's or commander's assistant. The agent assists the logistician to take appropriate action in the event of a logistics plan detected to be unsuccessful. In our case, a plan will be called as unsuccessful if it does not meet the timing requirements as specified in TPFDD (Time Phased Force Deployment Data). The agent is intended to be integrated with DIAS (Dynamic Information Architecture System, developed by Argonne National Lab, Chicago) and a customized Java based graphical user interface, together is called as DIAL. The agent interacts with the logistician and the integrated simulation system i.e. DIAS to arrive at an appropriate action. The complete system is summarized in figure 1.0.

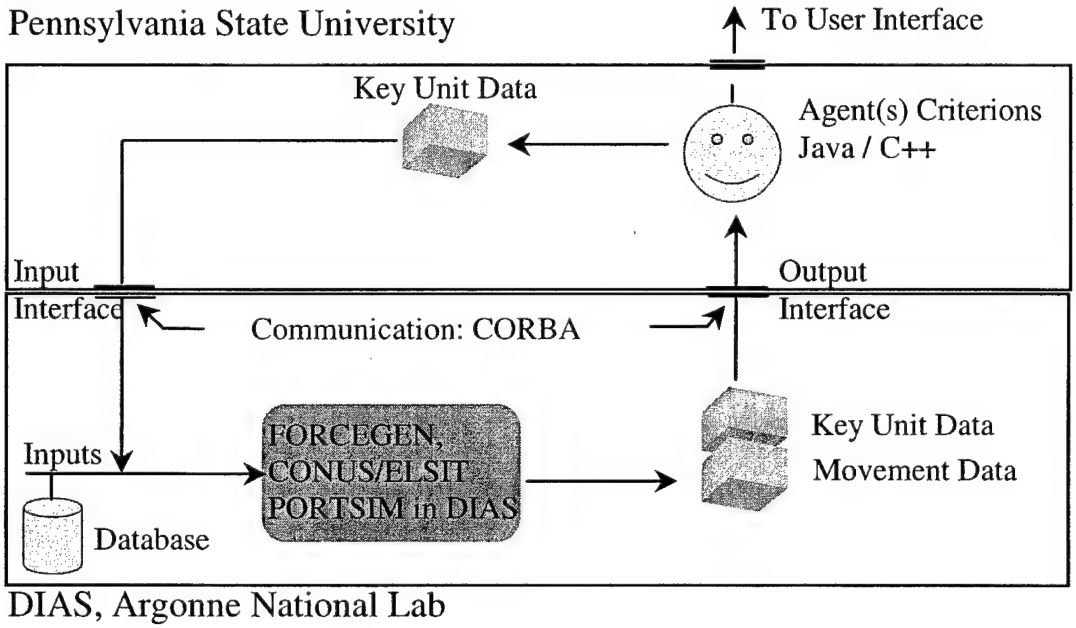


Figure1.0 DIAL architecture: DIAS, Agent and GUI

In figure 1.0, following steps are followed:

1. DIAS simulates end-to-end movement of equipment and forces through FORCEGEN, CONUS/ELIST and PORTSIM using (real-time and legacy) data.
2. DIAS creates an object from the subset of output data of simulations, called Key Unit Data (KUD) (and movement data). This object(s) is generic and independent of simulation programs.
3. DIAS sends the object to the agent through CORBA method invocation.
4. The agent uses KUD and movement data and its reasoning scheme to suggest an action or strategy, in case KUD indicates failure in maintaining TPFDD requirement. The agent also

suggests the parameters that need to be modified if the suggested strategy is applied. The suggested strategy is offered through a graphical user interface, built in Java.

5. The logistician may either accept or request for another alternative strategy.
6. After several interactions with the logistician, and upon acceptance of a strategy, the agent modifies the KUD based on the accepted strategy and sends it back to DIAS.
7. DIAS uses the modified KUD to simulate appropriate parts of the end-to-end movement.
8. The iterative process between DIAS and agent continues until the logistician approves the logistics plan.

In the light of the above iterative decision making procedure, the following are the key aspects in the development of agent and GUI by Pennsylvania State University.

1. An agent structure was developed which separates communication, message interpreting and reasoning as separate layer. This helps in developing different agents with different reasoning abilities, but with same communication and message understanding abilities.
2. Agent's strategies are implemented in the form of method invocations that use only KUD data and agent's past state to arrive at suggestive action. This form of implementation helps to add more strategies, though not dynamically.
3. Agent supports CORBA method invocation and string based communication (which is similar to KQML form of communication) between DIAS and GUI respectively.
4. GUI implementation in Java enables multiple logisticians to access DIAL simultaneously through Internet. GUI also lets the logistician to examine only relevant data rather than complete TPFDD, KUD, or simulation output.

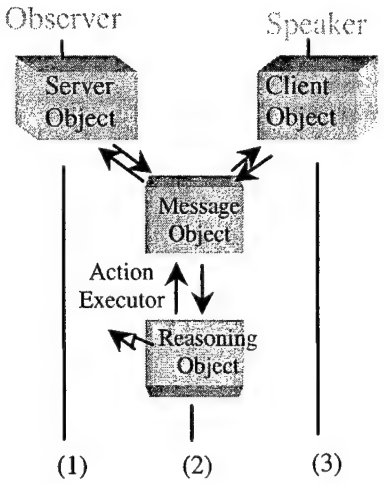
In the following sections, we shall briefly describe the structure of the agent, strategies, and GUI developed as a part of the current proposal.

2 Agent Structure

The agent is composed of three layers:

1. *Communication layer* consisting of **Server** object and **Client** object. Bot these objects supports socket-based communication and CORBA based method invocation or implementation.
2. *Message understanding layer* consisting of **Message** object. **Message** object contains procedures to parse strings received via socket and insert them to the knowledge base. The knowledge base in the case of agent is unit information, predicted timing information, the asset information, the state of the agent etc. **Message** object also contains methods to verify if the a string to be sent is of correct format. Finally, **Message** object also inherits CORBA object (or implementation stub) generated from IDL . The implementation stub contains procedures to extract CORBA defined data and maps them to knowledge base.

3. *Reasoning Layer* consists of various methods to process knowledge base and take appropriate action. The action in this case could be reply to a message, or send a request to another entity or agent, or access a data base etc. In our case, the external entities are DIAS and GUI.



The relationship between the objects are shown in the form of 'uses-a' relationship

Figure 2:0: Structure of the agent

Figure 2.0 shows the relationship between **Server** & **Client** objects, **Message** Object and **Reasoning** object. The agent structure is developed in the spirit of the agent being a persistent object, which can receive, process and send messages autonomously. However, from the implementation standpoint, it may happen that while processing a message, the agent may ignore receiving another message which may affect processing of the previous message. Similarly, processing of a message might delay the dispatching of another message which could not be accomplished before. Similarly, waiting for a message may delay certain reasoning task. Therefore, the agent must bear with task controller that must allocate certain time for receiving, processing and sending of messages. These three tasks are indicated as (1), (2) and (3) in figure 2.0. Currently, the task controller of the agent structure is implemented in the form of invoking these three tasks in a cyclical manner without allocating any time for each task.

2.1 Communication

The communication layer is composed of two objects. The role of **Server** object is to either receive messages from a socket or wait for CORBA method invocation. It implements both activities as non-blocking.

Message arriving at a socket:
If a message arrives at the socket (in our case, the message at the socket arrives from one of GUIs), the **Server** object passes the string to the **Message** object which contains methods to parse the string, extract relevant elements and insert them to the knowledge base.

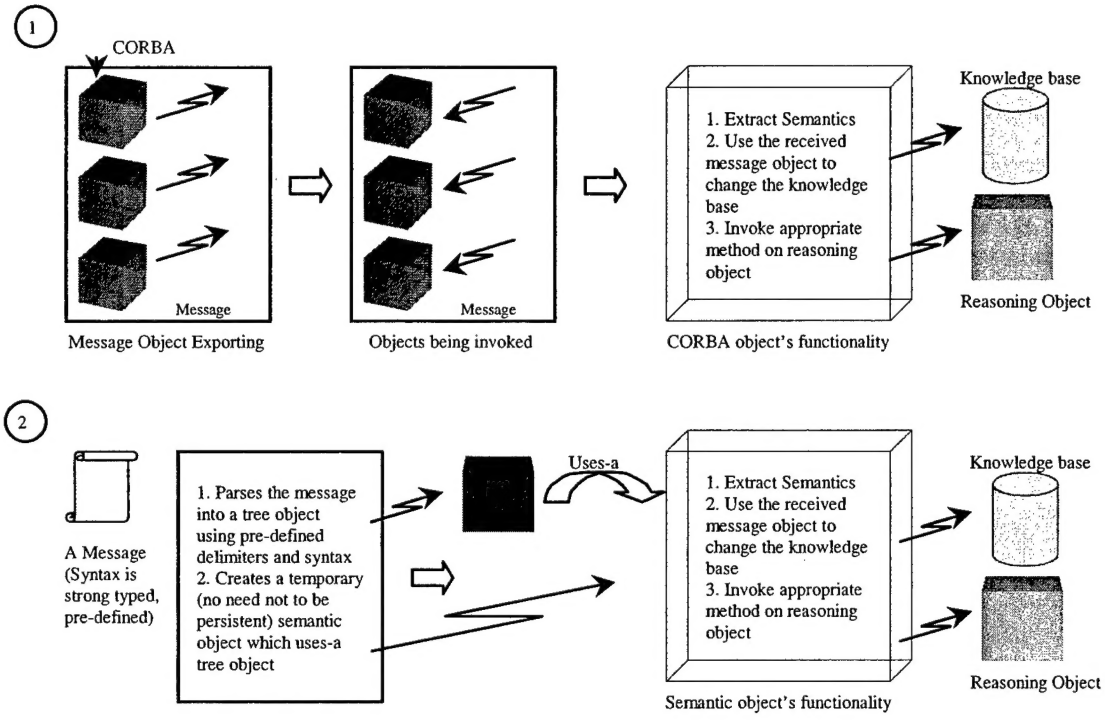


Figure 3.0 Message interpretation by the **Message** object

CORBA method invocation:

In order for another entity (say DIAS) to perform CORBA method invocation, the agent must have exported those methods. The **Server** object, which inherits the CORBA object generated from IDL, exports its methods. The underlying CORBA library while waiting for CORBA method invocation automatically invokes these methods. Waiting for CORBA method invocation is implemented in a non-blocking fashion.

The second object of the communication layer is **Client** object that allows the agent to send string messages and import CORBA methods. **Client** object uses methods **Message** object to check the parsing error in string message before it sends the message. The **Client** object also imports CORBA methods, which could be used by the **Reasoning** object. When the **Reasoning** object invokes the imported methods contained in **Client** object, the underlying CORBA library performs the request (i.e. method invocation) and delivery (i.e. return arguments of the method) of CORBA defined data from remote method implementation. The **Client** object uses methods of **Message** object to extract CORBA defined data and insert then in the knowledge base. In our case, since agent does not initiate any communication (because other external entities are not server), the Client object is not used in the current implementation of agent.

2.2 Message Interpretation

The **Message** object contains procedures for (1) parsing a string given a grammar, (2) extract the semantics of it and insert them into a given knowledge base, (3) extract CORBA defined data and insert them in the knowledge base. The procedures for parsing a string is usually generated using lexical analyzer and parsing tool such as flex and bison. However, in this implementation of agent we developed our own the parsing procedures (which efficient for small grammar, but may not be efficient for complicated grammar such as KQML, SQL etc.). The procedures scan the string and arrange the token/key (along with its semantics i.e. the value) in the form of a tree object. A tree object[1] is also created while extracting CORBA defined data. This tree object is used by another set of procedure to copy the data into the knowledge base. These methods are summarized in figure 3.0.

2.3 Reasoning

Reasoning object contains various methods mapped to process different types of messages. For example, when a string message arrives, the **Message** object invokes **Reasoning** object to 'act on the message', after it has parsed and inserted them in the knowledge base. The **Reasoning** object reads the knowledge base to determine which methods need to be invoked. In our case, these methods correspond to list of strategies. For example, a message concerning a strategy 'replacing of unit' will lead various methods such as (1) suggest replace units or (2) reverse replaced units or (3) analyze the impact of replacing units (which will lead to invocation of simulation in DIAS) etc. Similarly, when a CORBA method is invoked by another external entity (say DIAS), the **Message** object calls methods of **Reasoning** object, which invariantly leads to comparison of KUD received with existing KUD and presenting the difference to the logistician.

If we remove the communication and message layer, then the link between GUI and DIAS is illustrated in figure 4.0.

2.3.1 List of strategies

Following are the list of strategies implemented in the agent as logistician's assistant[2]

1. Suggest change of g-date that *may* make units arrive at theater that meet the TPFDD requirement.
2. Suggest replacement of similar units in terms of type of unit, passenger and equipment strength, (and mission and training level which we are not using currently) which *may* make units meet TPFDD requirement with little alteration in mission or unit.

---

[1] A tree object contains node and several child nodes. Similarly each child node branches out to several other child nodes. In our case, each contains a key and its value. For example, a key of a node can be 'assetInfo' and the value could be a complete string of asset information. Its child nodes will contain asset name, model and quantity information in separate nodes.

[2] It must be noted that agent only suggests the following actions. It does not implements or simulate the impact of these actions unless logistician approves them.

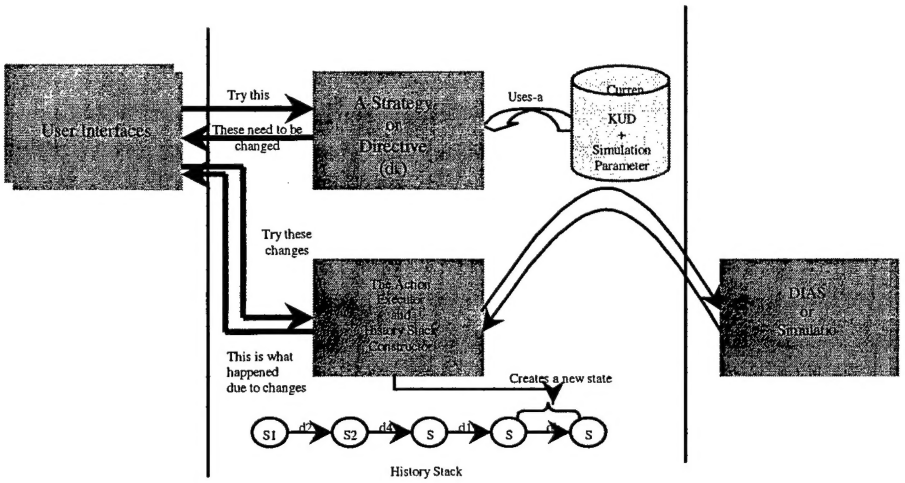


Figure 4.0 Interaction between GUI, **Reasoning** object and DIAS

3. Suggest addition of certain critical or bottleneck transportation assets, which *may* make movement of cargo faster.
4. Suggest acceptance of lateness of certain (less important) units and thereby changing TPFDD requirement, which *may* cause other units to arrive sooner.
5. Suggest abandoning mission of some units or halt deployment of certain units (which we call as deletion of units) which *may* cause deployment and movement of other units faster.

The impact of these suggestions is not guaranteed. However, it could be verified to certain extent through the usage of the simulation.

2.4 Other features

Various other features of the agent as logistician's assistant are (1) collaboration among multiple logisticians and (2) allowing the logisticians to backtrack certain strategies (i.e. strategy rollback scheme). We implement the collaboration in a primitive manner, where the agent sends notices to all the logisticians, if one of logistician makes change to the unit information that may cause strategy selection of rest of the logistician. We would like to implement the collaboration through conflict resolution based on hierarchy of the logisticians.

The rollback feature assumes that the result of the simulation is deterministic i.e. the simulation uses stochastic parameters, but the result of several repetition of simulation is same. Therefore, the agent stores only the type of strategy, rather than the results of applying the strategy (this improves inefficiency in terms of storing large amount of KUD information). Agent can reverse a strategy by applying the inverse of the strategy in the simulation.

3 Graphical User Interface (GUI)

Graphical user interface helps the logistician to view only relevant part of KUD, TPFDD or simulation results. The reader may access the web page - http://www.iddr.ie.psu.edu/~dial/Restricted to view current implementation of GUI. GUI presents list of units that are late when compared to TPFDD, along with the unit name and ULN information. GUI lets the logisticians to either chose a strategy or select the automatic mode where agent offers the current best strategy based on a primitive meta-reasoning scheme. The GUI is implemented in Java that enables logistician to connect to the agent via web. The GUI is fairly simple to use as has been reported from several past demonstrations to logisticians.

4 Future Work

The logistician's assisting agent offers an exhaustive list of embedded strategies to the logistician irrespective of the scenario (i.e. crisis type or combat type), sequence of past strategies and expected impact of the application of a strategy. It is observed that a logistician chooses only a specific set of strategies based on a scenario and the past choice of strategies. We propose learning schemes to be incorporated in the agent to enable it to learn logistician's choice of strategies and thus, offer only a subset of strategies. The learning schemes could be either supervised neural network model or a stochastic automaton model. For example, a neural network could be trained to learn the choice of logisticians' strategy based on following inputs:

1. Simulation input parameters such as resources and their status.
2. Simulation output parameters such as time of arrival and departure of units (Forces and equipment) at various nodes of a logistics graph.
3. TPFDD date (schedule) violations i.e. the delays of estimated dates from the scheduled dates.
4. The sequence of strategies that have been applied in the past.
5. Recommended modifications of simulation input parameters corresponding to each embedded strategy.
6. Scenario description.
7. Expected effect / impact of the modifications on the simulation output, if the recommended modifications are approved. This assumes that the agent bears a meta-model of simulations to determine expected effects.

A trained network will offer strategies that the logistician is most likely to accept.